1'P

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/664,418	09/17/2003	Bruce D. Holenstein	9203-27U1	3213

570 7590 06/07/2006

AKIN GUMP STRAUSS HAUER & FELD L.L.P.
ONE COMMERCE SQUARE
2005 MARKET STREET, SUITE 2200
PHILADELPHIA, PA 19103

EXAMINER	
HICKS, MICHAEL J	
ART UNIT	PAPER NUMBER
2165	

DATE MAILED: 06/07/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/664,418	Applicant(s) HOLENSTEIN ET AL.	
	Examiner Michael J. Hicks	Art Unit 2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 March 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 September 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-9 are pending in the instant application.

Response to Arguments

2. Applicant's arguments with respect to the rejections of Claims 1, 2, 4 under 102(e) in view of Bortvedt et al. and Claims 8-9 under 102(b) in view of Holliday et al. have been fully considered and are persuasive. Therefore, the rejections have been withdrawn. However, upon further consideration, a new grounds of rejection is made in view of Son et al.

3. Applicant's arguments in regards to Claims 6-7 have been fully considered but they are not persuasive.

Applicant argues that Holliday does not teach performing the dual write replication upon the occurrence of an update step or operation because Holliday states that update operations are delayed until commit time. Examiner asserts that when a write or update *command* is received at a database, no data is actually updated until the commit time of the operation. Therefore, an update *step or operation* (e.g. the actual writing of the data) does not occur until the commit time of the update command. In this respect, Holliday does teach that the dual write replication is performed upon the occurrence of an update step or operation (e.g. the commit time of an update command) and the rejection is maintained.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1, 2, 4, and 8 rejected under 35 U.S.C. 102(b) as being anticipated by Son et al. ("Replication Control for Distributed real-Time Database Systems", In Proc. of Int. Conf. on Distributed Computing Systems, pages 144--151, 1992; IEEE and referred to hereinafter as Son).

As per Claim 1, Son discloses a method of replicating data associated with a plurality of transactions in a replication system including a plurality of nodes connected via communication media in a topology, each node including a database (i.e. *"A distributed system consists of multiple autonomous computer systems (sites) connected via a communications network...We assume that the database is fully replicated at all sites. Read and Write are the two fundamental types of logical operations that are implemented by executing the respective physical operation on one or more copies of the physical data object in question."* The preceding text excerpt clearly indicates that a method of replicating data associated with a plurality of transactions (e.g. read and write transactions) in a replication system (e.g. replicated database) including a plurality of nodes (e.g. sites) connected via a media topology (e.g. communications network) with each node containing a local database exists.) (Page 145, Column 1, Paragraph 3), at least some of the nodes being able to independently receive and post transactions (i.e. *"We assume that the database is fully replicated at all sites. Read and Write are the two fundamental types of logical operations that are*

implemented by executing the respective physical operation on one or more copies of the physical data object in question." The preceding text excerpt clearly indicates that the sites (e.g. nodes) are able to receive and post transactions at their local databases independently.) (Page 145, Column 1, Paragraph 3), the method comprising: (a) initiating and performing transactions to be executed in a database at an originating node (i.e. *"Two types of transactions are allowed in our environment: query transactions and update transactions. Query transactions consist of only read operations that access data objects and return their values to the user. Update transactions consist of both read and write operations. They execute a sequence of local computations and update the values of all replicas of each associated data object."* The preceding text excerpt clearly indicates that transactions are received initiated and performed at the local/originating node.) (Page 145, Column 2, Paragraph 2); and (b) replicating the transactions to at least one or more other nodes by: (i) pausing each transaction being executed in the database at the originating node prior to a commit operation for the transaction (i.e. *"Update transactions commit by employing a two phase protocol. In the first phase (vote phase), an update transaction sends an update message to each token-site of every data object in the write set. The transaction waits until it gets a response from all token-sites for each data object."* The preceding text excerpt clearly indicates that the transaction is paused in order to wait for a response to the update message before the transaction commits.) (Page 146, Column 1, Paragraph 6), (ii) assigning a ready to commit token to the transaction (i.e. *"In the first phase (vote phase), an update transaction sends an update message to each token-site of every data object in the write set. The transaction waits until it gets a response from all token-sites for each data object."* The preceding text excerpt clearly indicates that a ready to commit token (e.g. an update message) is assigned to the transaction.) (Page 146, Column 1, Paragraph 6), (iii) sending the ready to commit token to the one or more other nodes (i.e. *"In the first phase (vote phase), an update transaction sends an update message to each token-site of every data object in the write set."* The preceding text excerpt clearly indicates that the update message/ready to commit token is sent to other nodes/sites.)

(Page 146, Column 1, Paragraph 6), (iv) determining at the one or more other nodes whether the respective databases are prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the originating node (i.e. *"The transaction waits until it gets a response from all the token-sites for each data object. If all token sites vote YES, then the transaction enters the second phase (commit phase)."* The preceding text excerpt clearly indicates that a ready to commit message/token is sent back in response to the update message/ready to commit token if the other nodes/sites are prepared for a commit operation.) (Page 146, Column 1, Paragraph 6), and (v) executing a commit operation at the database of the originating node only upon receipt from at least one of the other nodes of the ready to commit tokens originally sent from the originating node (i.e. *"If all token sites vote YES, then the transaction enters the second phase (commit phase). It sends the actual value of each data object to be written to the respective token sites."* The preceding text excerpt clearly indicates that after a response from the one or more other nodes/sites the operation commits.) (Page 146, Column 1, Paragraph 6), wherein step (b) is performed asynchronously with respect to other transactions that are initiated in step (a) (i.e. *"Transactions arriving at the system are assumed to be non-periodic...Epsilon-serializability (ESR) is a correctness criterion that enables asynchronous maintenance of mutual consistency of replicated data."* The preceding text excerpt clearly indicates that the operations arrive and are replicated asynchronously, and that the replication is done asynchronous to the arrival of operations.) (Page 145, Column 2, Paragraph 3; Page 146, Column 2, Paragraph 5).

As per Claim 2, Son further discloses the commit operation in step (b)(v) is performed only upon receipt from each of the one or more the other nodes of the ready to commit tokens originally sent from the originating node (i.e. *"If all token sites vote YES,*

then the transaction enters the second phase (commit phase). It sends the actual value of each data object to be written to the respective token sites." The preceding text excerpt clearly indicates that after a positive response from each of the one or more other nodes/sites the operation commits.) (Page 146, Column 1, Paragraph 6).

As per Claim 4, Son discloses a method of replicating data associated with a plurality of transactions in a replication system including a plurality of nodes connected via communication media in a topology each node including (i) a database, (ii) a replication engine which performs data replication functions (i.e. *"A distributed system consists of multiple autonomous computer systems (sites) connected via a communications network...We assume that the database is fully replicated at all sites. Read and Write are the two fundamental types of logical operations that are implemented by executing the respective physical operation on one or more copies of the physical data object in question."* The preceding text excerpt clearly indicates that a method of replicating data associated with a plurality of transactions (e.g. read and write transactions) in a replication system (e.g. replicated database) including a plurality of nodes (e.g. sites) connected via a media topology (e.g. communications network) with each node containing a local database exists.) (Page 145, Column 1, Paragraph 3), and (iii) an application which executes transactions and posts the transactions to the database, the application being independent of the replication engine, each transaction being one or more transaction steps or transaction operations (i.e. *"Two types of transactions are allowed in our environment: query transactions and update transactions. Query transactions consist of only read operations that access data objects and return their values to the user. Update transactions consist of both read and write operations. They execute a sequence of local computations and update the values of all replicas of each associated data object."* The preceding text excerpt clearly indicates that transactions are received initiated and performed at the local/originating node independently of replication, and that each transaction is one or more transaction

steps or operations (e.g. a sequence of local computations).) (Page 145, Column 2, Paragraph 2), the method comprising: (a) an application at a first node that initiating and performs transactions to be executed in a database at an originating node (i.e. *"Two types of transactions are allowed in our environment: query transactions and update transactions. Query transactions consist of only read operations that access data objects and return their values to the user. Update transactions consist of both read and write operations. They execute a sequence of local computations and update the values of all replicas of each associated data object."* The preceding text excerpt clearly indicates that transactions are received initiated and performed at the local/originating node.) (Page 145, Column 2, Paragraph 2), the application pausing each transaction being executed in a source database at the first node prior to a commit operation for the transaction (i.e. *"Update transactions commit by employing a two phase protocol. In the first phase (vote phase), an update transaction sends an update message to each token-site of every data object in the write set. The transaction waits until it gets a response from all token-sites for each data object."* The preceding text excerpt clearly indicates that the transaction is paused in order to wait for a response to the update message before the transaction commits.) (Page 146, Column 1, Paragraph 6); (b) a replication engine at the first node assigning a ready to commit token to the transaction in coordination with the application (i.e. *"In the first phase (vote phase), an update transaction sends an update message to each token-site of every data object in the write set. The transaction waits until it gets a response from all token-sites for each data object."* The preceding text excerpt clearly indicates that a ready to commit token (e.g. an update message) is assigned to the transaction.) (Page 146, Column 1, Paragraph 6); (c) the replication engine at the first node sending the ready to commit token to the second node (i.e. *"In the first phase (vote phase), an update transaction sends an update message to each token-site of every data object in the write set."* The preceding text excerpt clearly indicates that the update message/ready to commit token is sent to other nodes/sites.) (Page 146,

Column 1, Paragraph 6); (d) a replication engine at a second node determining whether a target database at the second node is prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the first node (i.e. *"The transaction waits until it gets a response from all the token-sites for each data object. If all token sites vote YES, then the transaction enters the second phase (commit phase)."* The preceding text excerpt clearly indicates that a ready to commit message/token is sent back in response to the update message/ready to commit token if the other nodes/sites are prepared for a commit operation.) (Page 146, Column 1, Paragraph 6); and (e) the application at the first node executing a commit operation at the source database in coordination with the replication engine only upon receipt from the second node of the ready to commit token originally sent from the first node (i.e. *"If all token sites vote YES, then the transaction enters the second phase (commit phase). It sends the actual value of each data object to be written to the respective token sites."* The preceding text excerpt clearly indicates that after a response from the one or more other nodes/sites the operation commits.) (Page 146, Column 1, Paragraph 6), wherein the replication engine operates asynchronously with respect to the application until step (b) occurs (i.e. *"Transactions arriving at the system are assumed to be non-periodic...Epsilon-serializability (ESR) is a correctness criterion that enables asynchronous maintenance of mutual consistency of replicated data."* The preceding text excerpt clearly indicates that the operations arrive and are replicated asynchronously, and that the replication is done asynchronous to the arrival of operations.) (Page 145, Column 2, Paragraph 3; Page 146, Column 2, Paragraph 5).

As per Claim 8, Son discloses a method of performing dual writes for replicating transactions among plural databases located at different nodes (i.e. *"A distributed system consists of multiple autonomous computer systems (sites) connected via a communications network...We*

assume that the database is fully replicated at all sites. Read and Write are the two fundamental types of logical operations that are implemented by executing the respective physical operation on one or more copies of the physical data object in question." The preceding text excerpt clearly indicates that a method of replicating data associated with a plurality of transactions (e.g. read and write transactions) in a replication system (e.g. replicated database) including a plurality of nodes (e.g. sites) connected via a media topology (e.g. communications network) with each node containing a local database exists.) (Page 145, Column 1, Paragraph 3), each transaction being one or more transaction steps or transaction operations (i.e. *"Two types of transactions are allowed in our environment: query transactions and update transactions. Query transactions consist of only read operations that access data objects and return their values to the user. Update transactions consist of both read and write operations. They execute a sequence of local computations and update the values of all replicas of each associated data object."* The preceding text excerpt clearly indicates that each transaction consists of one or more transaction steps or transaction operations (e.g. a sequence of local computations).) (Page 145, Column 2, Paragraph 2), at least some of the transaction steps or transaction operations being update steps or operations which are performed only after database locking occurs (i.e. *"Transaction T2 requests to read a data object X for which transaction T1 has already issued a write request...If T2 is younger that T1 then the original token-based scheme requires that T2 must wait for the termination of T1 before it reads the value of X."* The preceding text excerpt clearly indicates that the transactions may be commit steps of a write instruction which holds a lock on the data object it is writing (e.g. read and other write operations must wair for the completion of the current write operation before they can be initiated).) (Page 149, Column 1, Paragraphs 4-5, Column 2, Paragraph 1), the method comprising: (a) initiating a transaction at an originating node (i.e. *"Two types of transactions are allowed in our environment: query transactions and update transactions. Query transactions consist of only read operations that access data objects and return their values to the user. Update transactions consist of both read and write operations. They execute a sequence of local computations and update the values of all replicas of each associated data object."* The preceding text

excerpt clearly indicates that transactions are received initiated and performed at the local/originating node.) (Page 145, Column 2, Paragraph 2); (b) the dual write replication process causing database locking to occur at one or more other nodes only upon the occurrence of an update step or operation in the transaction at the originating node. (i.e. *"Update transactions commit by employing a two phase protocol. In the first phase (vote phase), an update transaction sends an update message to each token-site of every data object in the write set. The transaction waits until it gets a response from all token-sites for each data object. If all token sites vote YES, then the transaction enters the second phase (commit phase). It sends the actual value of each data object to be written to the respective token sites."* The preceding text excerpt along with the above sited excerpts clearly indicates that the dual write replication process causes database locking to occur at one or more other nodes only during an update step (e.g. commitment of an update transaction) at the originating node.) (Page 146, Column 1, Paragraph 6).

6. Claims 6-7 rejected under 35 U.S.C. 102(b) as being anticipated by Holliday et al. ("The performance of database replication with group multicast", Fault-Tolerant Computing, 1999. Digest of Papers. Twenty-Ninth Annual International Symposium on 15-18 June 1999 Page(s):158 – 165 and reffered to hereinafter as Holliday).

As per Claim 6, Holliday discloses a method of performing dual writes for replicating transactions among plural databases located at different nodes (i.e. *"We therefore consider replicating several copies of the database (or perhaps only the hot-spot pages) on multiple sites connected by a local area network...we assume that concurrency control is locally enforced by strict two phase locking at all server sites."* The preceding text excerpt clearly indicates that a replicated database with database copies located at different sites/nodes exists which utilized two-phase locking/dual writes.) (Page 159, Column 2, Paragraph 4; Page 159, Column 1, Paragraphs 1-2), each

transaction being one or more transaction steps or transaction operations (i.e. *"This is achieved by deferring update operations until commit time, when a single message with all updates is sent to all other sites."* The preceding text excerpt clearly indicates that each transaction is a collection of/one or more read and write operations (e.g. transaction operations.) (Page 160, Column 1, Paragraph 2), at least some of the transaction steps or transaction operations being update steps or operations (i.e. *"This is achieved by deferring update operations until commit time, when a single message with all updates is sent to all other sites."* The preceding text excerpt clearly indicates that at least some of the transaction operations are write operations (e.g. update operations).) (Page 160, Column 1, Paragraph 2), the method comprising: (a) initiating a transaction at an originating node (i.e. *"A transaction Ti, executes a read operation locally, while a write operation is deferred until Ti is ready to commit."* The preceding text excerpt clearly indicates that the collection of read and write operations/transaction is initiated at an originating node.) (Page 160, Column 1, Paragraph 2); (b) inhibiting the dual write replication process from communicating transaction steps or operations of the transaction with one or more other nodes until an update step or operation occurs within the transaction at the originating node (i.e. *"A transaction Ti, executes a read operation locally, while a write operation is deferred until Ti is ready to commit. To terminate, Ti broadcasts its deferred writes to all sites."* The preceding text excerpt clearly indicates that the broadcast of write operations/beginning of the dual write replication process is delayed until the write/update operations occur.) (Page 160, Column 1, Paragraph 2); and (c) upon the occurrence of the update step or operation, performing the dual write replication process with respect to the one or more other nodes and sending with the update step or operation all transaction steps or operations for the transaction that have occurred prior to the update step or operation for the transaction, or prior to the previous update step or operation if a previous update step or operation existed for the transaction (i.e. *"To*

terminate, Ti broadcasts it's deferred writes to all sites. On receiving the writes, the lock manager on site S grants all write locks to Ti atomically, and then the writes are executed at S. After all the writes of Ti are executed locally, Ti broadcasts its commit operation to all sites. Ti terminates after the delivery and execution of its commit locally." The preceding text excerpt clearly indicates that the two phase lock operation/dual write replication process is executed with respect to the other sites/one or more other nodes, and that all writes/transaction operations which have occurred prior to the update operation are sent during the execution.) (Page 160, Column 1, Paragraph 2).

As per Claim 7, Holliday further discloses (d) determining if the originating node needs to receive a data record from the one or more other nodes during the dual write replication process (i.e. *"To terminate, Ti broadcasts its deferred writes to all sites."* The preceding text excerpt clearly indicates that because the update operations are write operations, the one or more other sites/nodes will always need to receive a data record.) (Page 160, Column 1, Paragraph 2); and (e) sending the data record to the originating node only if it is determined that the originating node needs the data record (i.e. *"To terminate, Ti broadcasts its deferred writes to all sites...After all writes of Ti are executed locally, Ti broadcasts it's commit operation to all sites."* The preceding text excerpt clearly indicates that if the one or more other sites/nodes are to commit a write operation, the record that was needed for the write operation must have been sent in the broadcast from the originating site/node.) (Page 160, Column 1, Paragraph 2).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and

> the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 3 and 5 rejected under 35 U.S.C. 103(a) as being unpatentable over Son in view of Arevalo et al. ("Deterministic Scheduling for Transactional Multithreaded Replicas", 19th IEEE Symposium on Reliable Distributed Systems (SRDS'00) p. 164, 2000 and referred to hereinafter as Arevalo).

As per Claims 3 and 5, Son fails to disclose the transactions are multi-threaded.

Arevalo discloses the transactions are multi-threaded (i.e. *"In this paper, we present a deterministic scheduling algorithm for multithreaded replicas in a transactional framework."* The preceding text excerpt clearly indicates that replicas are made through transactions in a multi-threaded manner.) (Page 164, Abstract).

It would have been obvious to one skilled in the art at the time of Applicants invention to combine the teachings of Son with the teachings of Arevalo to include the transactions are initiated and performed in a multi-threaded manner with the motivation to be able to execute several transactions concurrently.

9. Claim 9 rejected under 35 U.S.C. 103(a) as being unpatentable over Son in view of Holliday.

As per Claim 9, Son fails to disclose (d) determining if the originating node needs to receive a data record from the one or more other nodes during the dual write

replication process; and (e) sending the data record to the originating node only if it is determined that the originating node needs the data record.

Holliday discloses (d) determining if the originating node needs to receive a data record from the one or more other nodes during the dual write replication process (i.e. *"To terminate, Ti broadcasts its deferred writes to all sites."* The preceding text excerpt clearly indicates that because the update operations are write operations, the one or more other sites/nodes will always need to receive a data record.) (Page 160, Column 1, Paragraph 2); and (e) sending the data record to the originating node only if it is determined that the originating node needs the data record (i.e. *"To terminate, Ti broadcasts its deferred writes to all sites...After all writes of Ti are executed locally, Ti broadcasts it's commit operation to all sites."* The preceding text excerpt clearly indicates that if the one or more other sites/nodes are to commit a write operation, the record that was needed for the write operation must have been sent in the broadcast from the originating site/node.) (Page 160, Column 1, Paragraph 2).

It would have been obvious to one skilled in the art at the time of Applicants invention to combine the teachings of Son with the teachings of Holliday to include (d) determining if the originating node needs to receive a data record from the one or more other nodes during the dual write replication process; and (e) sending the data record to the originating node only if it is determined that the originating node needs the data record with the motivation to take advantage of atomic broadcast systems in distributed replicated databases to simplify message passing and conflict resolution in hopes of making replication efficient.

Points of Contact

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Hicks whose telephone number is (571) 272-2670. The examiner can normally be reached on Monday - Friday 8:30a - 5:00p.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on (571) 272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael J Hicks
Art Unit 2165
(571) 272-2670

Apu Mofiz
Primary Examiner
TC 2100